

```

#####
#
LocationToBinary <- function(n){
  binary.vector <- rev(as.numeric(intToBits(n)))
  first.1 <- (which(binary.vector == 1)[1])
  binary.vector <- binary.vector[-(1:(first.1-1))]
  return(binary.vector)
}
#
#####
#
# system = 1 Stern-Brocot
# system = 2 Bird
# system = 3 HCS
# system = 4 Yurramendi-2
#
BinaryToCF.2.1 <- function(binary.vector){
  nbin <- binary.vector
  nbin[length(nbin) + 1] <- 2
  dnbin <- nbin[-length(nbin)] - nbin[-1]
  cuts <- which(dnbin != 0)
  runlengths <- c(cuts[1], cuts[-1] - cuts[-length(cuts)])
  cf <- runlengths
  cf[1] <- cf[1] - 1
  cf[length(cf)] <- cf[length(cf)] + 1
  return(cf)
}
#
BinaryToCF.2.2 <- function(binary.vector){
  filter <- rep(c(0,1), 16)[1:length(binary.vector)]
  nbin <- as.numeric(binary.vector == filter)
  nbin[length(nbin) + 1] <- 1
  nbin <- nbin[-1]
  dnbin <- c(nbin[-length(nbin)] - nbin[-1], 2)
  cuts <- which(dnbin != 0)
  runlengths <- c(cuts[1], cuts[-1] - cuts[-length(cuts)])
  cf <- runlengths
  if(length(binary.vector) > 1 & binary.vector[2] == 0) cf <- c(0, cf)
  return(cf)
}
#
BinaryToCF.2.3 <- function(binary.vector){
  nbin <- binary.vector
  nbin <- rev(c(nbin, 0, 0))
  ones <- rev(which(nbin == 1) - 1)
  dones <- c(ones - c(ones[-1], 0))
  if(length(ones) == 1) dones <- ones
  #
  dones[1] <- dones[1] - 1
  if(dones[1] == 0) dones <- dones[-1]
  cf <- dones
  if(length(binary.vector) > 1 & binary.vector[2] == 0) cf <- c(0, cf)
  return(cf)
}
#
BinaryToCF.2.4 <- function(binary.vector){
  nbin <- binary.vector
  zeros <- c(which(nbin == 0), length(nbin))
  dzeros <- zeros[-1] - zeros[-length(zeros)]
  cf <- c(zeros[1] - 2, dzeros[-length(dzeros)], dzeros[length(dzeros)] + 2)
  if(sum(which(nbin == 0)) == 0) cf <- length(nbin)
  return(cf)
}
#
#####
#
CFToFraction <- function(cf){
  r <- cf[1]
  if(length(cf) > 1){
    r <- 1/cf[length(cf)]
    for(i in 1:(length(cf)-1)) r[i+1] <- 1/(cf[length(cf)-i] + r[i])
    r[length(cf)] <- cf[1] + r[length(cf)-1]
  }
  return(fractions(r[length(r)]))
}
#
#####
#
LocationToFraction.CF.2 <- function(system, n){
  #
  # system = 1 Stern-Brocot
  # system = 2 Bird
  # system = 3 HCS
  # system = 4 Yurramendi-2
  #
  # Location --> Binary --> Continued Fraction --> Fraction
  #
  # Location --> Binary
  #
  binary.vector <- LocationToBinary(n)
  #
  # Binary --> Continued Fraction
  #
  if(system == 1) cf <- BinaryToCF.2.1(binary.vector)
  if(system == 2) cf <- BinaryToCF.2.2(binary.vector)
  if(system == 3) cf <- BinaryToCF.2.3(binary.vector)
  if(system == 4) cf <- BinaryToCF.2.4(binary.vector)
  #
  # Continued Fraction --> Fraction
  #
  fraction <- CFToFraction(cf)
  #
  return(fraction)
}
#
#####
#
# Examples: Compute fraction of the 47th in all four systems
#
library(MASS)

```

```
#
LocationToFraction.CF.2(1, 47) # 5/ 6
LocationToFraction.CF.2(2, 47) # 5/13
LocationToFraction.CF.2(3, 47) # 8/13
LocationToFraction.CF.2(4, 47) # 1/ 6
#
#####
```